APPLICATION NOTE AN-C8805-1 Issue 1 May 1988

CONVERSION OF TAPE CARD TO GENERAL RS-232 USE

Introduction

The Dual Tape Interface Card, "DTI-1", for Interak was developed in 1982 and before the advent of floppy disks it was a vital part of an Interak system. In accordance with the principles of Interak design it has proved to be long lasting and it is still in production for those who want it, however the current typical user of Interak has moved over entirely to floppy disk storage in place of audio cassette tape.

A number of users have expressed a desire for a simple RS-232 serial interface. (ie an RS-232 interface which is easy for a "simple minded" user to understand; the current quad serial interface board QS-1 for Interak with its software selectable baud rate, modem control lines, etc, etc, is often too complicated for a simple task of say adding a serial terminal, or a serial printer to a current Interak CP/M Plus system.

One outstanding benefit of doing things the simple way is that the existing CP/M Plus implementation already contains serial routines to allow its use with a serial terminal and/or a serial printer, so the user does not have to write new software to use say a serial printer interface in place of the parallel printer interface (usually PRN-3).

The modification requires the installation of 2 ICs on the patch areas provided, a number of discrete components, a few track cuts and some point to point wiring.

Features of Modified Card

RS-232 level output RXD (Received Data), conventionally connected to pin 3 of a 25-way "D" type connector.

RS-232 level input TXD (Transmitted Data), conventionally connected to pin 2 of a 25-way "D" type connector.

RS-232 level input DTR (Data Terminal Ready), conventionally connected to pin 20 of a 25-way "D" type connector. This is used as a handshake signal to regulate the flow of data from this card to the receiving device. It has particular relevance to the use of the card as a serial printer interface, where this signal could then be called a "busy" signal. (Depending on the whim of the printer manufacturer it may be found on some other pin than pin 20.)

Front panel Selection of baud rates:

300	baud	transmit and receive
1200	baud	transmit and receive
9600	baud	transmit and receive

Availability (with further modification) of baud rates:

75	baud	transmit and receive
150	baud	transmit and receive
300	baud	transmit and receive
600	baud	transmit and receive
1200	baud	transmit and receive
2400	baud	transmit and receive
4800	baud	transmit and receive
9600	baud	transmit and receive
19200	baud	transmit and receive

Description of Modification Circuit

The modification circuit is simply a copy of the relevant page of the DTI-1 Card circuit diagram; the one which shows U15, the 1015 UART ("UART" = "Universal Asynchronous Receiver Transmitter"). (In the original DTI-1 notes this diagram was the one on page 17).

At the top of the UART diagram you can see the original Serial In and Serial Out signals SI and SO, pins 20 and 25 respectively. These used to go to other circuitry on the DTI-1 card which made the conversion to and from audio tones to record on a tape cassette, but now they are taken via appropriate sections of UX1, a 14506 RS-232 driver receiver. The 14506 is better in this application than the well known 1488 and 1489 driver and receiver ICs, because both driver and receiver sections are contained in the one IC, making the construction task far easier, as we only have space for one IC for this purpose in this modification.

Note that this interface receives on the interface transmit line and transmits on the interface receive line - this suits most of the peripherals which may be connected to this interface. It's an odd world outside Interak, but we have to put up with it!

In the bottom right hand corner of the diagram is another section of the 14506, a receiver for the printer "busy" signal. (We are referring to a printer for convenience of description, but of course devices other than printers can be connected to the finished interface.) Often the busy signal from a serial printer is connected to pin 20 of the RS-232 25-way "D" type connector, which is the usual pin for the signal known as DTR (data terminal ready). The destiny of the busy signal will be revealed in a moment; meanwhile we direct your attention to the last part of the 14506 discussion. The power supplies are given in the little box towards the bottom right hand corner of the diagram. Diodes have been added to protect the 14506 against certain

external adverse conditions, but they are not essential and with a bit of luck no harm will result if the +12V, -12V, ØV, and +5V supplies are taken directly to the power pins of the IC. Decoupling capacitors are desirable here too as a matter of style, but we think it will be safe enough to omit them if space and patience in construction ~~is~~ are at a premium.

Because the 145406 is a CMOS device the unused logic signal inputs pins 10 and 14 are not left floating. There is no need to terminate pin 2, the RS-232 receiver, because of course the chip designers have to make it safe to leave an RS-232 line open, since this happens regularly in normal use of an RS-232 system.

We shall now explain the circuit connected to the "Busy" ("DTR") signal. Remember that the flow of serial data out of a UART is basically regulated by the TBMT (Transmit Buffer Empty) signal. Before the computer sends data it should check that the buffer is empty. This is done by "reading the status" of the UART, and examining the top bit, D7, of the "status word", (TBMT is routed to D7 in this design). Of course whether or not the UART can cope is not the whole story. The peripheral device may be in difficulty, for example if it is a printer, being sent data at a rate greater than the rate at which it can print.

To save characters being lost the printer "busy" signal (for example) is brought into the gates of UX2, a 74LS33 quad 2-input NOR IC, and overrules the empty status the UART may be reporting. The UART may have an empty buffer, but if the printer is busy it will fool the software into thinking that the UART transmit buffer is not empty, ie D7 will be held low when the status word is read from the UART.

You may think this is a bit fishy, because under adverse conditions the software may loop for ever waiting for example for a printer which has been switched off, and you would be right. If so you are inclining now towards the philosophy behind the design of the Interak QS-1, Quad Serial Card, and matters become far more complicated. We are now in the happy position of being able to answer all criticism - if you think this simple design is too naive you can be directed to the highly sophisticated "QS-1", and if you think the QS-1 is over complicated you can be directed back here.

An open collector device has been used for UX2 because D7 is part of the internal data bus of the card and cannot be connected to a conventional gate, which is always "1" or "Ø" at its output and does not allow the opposite state without an argument. The conventional way of dealing with such matters is to use a "tristate" IC, but in this modification we cannot afford the space for another IC.

The output of an open collector IC is effectively "off" (tristate) when its input conditions are such as to cause the logic "1" state at the output. (This is because inside an open collector gate there is only one active device,

which pulls down to a logic "Ø", and there is effectively no connection whatever at the output when the pull down device is off.) When the internal bus is required for the flow of data to and from the UART the signal at pin 8 of UX2 is "1" and the invitable action of a NOR gate is that pin 1Ø is thus "Ø", regardless of the state of pin 9. Thus output pin 13 of UX2 is "1" (or more correctly off altogether) and it does not affect the data line D7 in any way.

However when the "read status" function is in operation pin 8 of UX2 is "Ø". Now the state of pin 1Ø (and hence pin 13 and D7) does depend of conditions at pin 9. Pin 9 of UX2 is "1" only when both TBMT is "1" and the printer is "not busy", ie the printer busy input is holding pin 2 of UX2 at "Ø". Under these conditions the software is safe to send data to the UART, but it must stop when either TBMT goes to a "1" or the busy signal at pin 2 of UX2 goes to a "1" (or both). The NOR gate action satisfies this requirement, which is why it is there of course.

One final point not mentioned at all is the purpose of the 4 pull up resistors RX1-RX4. These are necessary to establish a logic "1" at the outputs of the open collector gates in UX2. As already explained, an open collector gate has no means of driving its output high, it can only "not drive it low". The resistance of RX1-4 is set at 3k3 (3.3 kilohms) which is small enough to ensure an open collector output rises smartly to a logic "1" when it is not "Ø", but not so small as to cause any problem to the other devices which may want to pull the logic level back down to "Ø" themselves.

The remaining parts of the modification are more easily described. We have assumed that the same baud rate will be used for both transmit and receive. The receive and transmit baud rates are set by the frequency at RCP and TCP (pins 17 and 4Ø respectively on the UART U15). The modification calls for these to be connected together. They were separate on the DTI-1 design, not intrinsically to allow different receive and transmit baud rates, but to allow fine tuning of the receive clock by a "phase locked loop" circuit to get good results from poor tape recorders. "Wow" and "flutter" and "dropouts" are not expected in the modern computer world, so the direct connection RCP to TCP is the most appropriate for the present application.

The design of the DTI-1 provided for front panel switch selection of 3 baud rates, 3ØØ, 12ØØ, 24ØØ, with 6ØØ as an option. These satisfy all the needs of the standard "CUTS" or "Kansas City" tape transfers ("CUTS" = "Cassette Users' Tape Standard" agreed one day in Kansas City), but they are not so suitable for other serial peripherals.

Arbitrarily we have chosen that after modification the rates selectable from the front panel will be as follows: 3ØØ, 12ØØ, 96ØØ. This means only a very slight alteration. The fastest of the three available baud rates is set by the frequency of the signal at pin 3 of U8. Originally it was 38.4 kHz, representing 24ØØ baud, so by connecting to a

signal of 153.6 kHz, four times higher, the 2400 baud rate is increased to 9600 baud.

The user can have any other of the available baud rates because they are all produced by a chain of binary dividers from the master crystal oscillator on the board.

This concludes the circuit description of the modifications.

Constructional Notes

The modification is as follows. We shall assume that an existing DTI-1 card is to be modified, rather than a bare DTI-1 being built from scratch to this modification.

1. Remove Front Panel, and all connections to the front panel components. (But if it suits you better leave FPS1 still connected to the board as it will be used in the finished job. Also if you wish you can retain the circuits for the tape motor control relays; they may have some use for controlling RS-232 switching for some special application.) In what follows we shall leave it to your own judgement to decide it your purpose is best served by keeping something we suggest you discard.

2. Remove the following components if you wish:

   All Jumper Links (JLinks)
   U1
   U7
   U13
   U14
   U17
   U18
   RLY1
   RLY2

3. Consult diagram "DTI-1 Alteratioins for ANC88Ø5, Layout of Modifications"

   Install a 16 pin DIL socket on Patch Area 1, and a 14 pin DIL socket on Patch Area 2. Only solder a few pins to keep the sockets in position at this stage, and note that the socket on Patch Area 2 is to be positioned to leave spaces for the resistors and wires to be fitted in the upper holes of the Patch Area.

4. Fit the discrete components CRX1,2,3 (assuming that you are using these, you may prefer to omit them), and RX1,2,3,4, all at your own pace. Note that many of the connections on the patch area can be made much more conveniently if you use the leads of the discrete components to join point to point before you solder and crop them.

5. There are 3 places where track cuts are to be made, and the diagrams show these and the other wiring which is necessary. Don't forget the links to be added, one near U9 and U1Ø, and the other near U11 and U15. (RHS of diagram).

   A useful type of wire is miniature prestripped "wire wrapping" wire, particularly when it comes to the connections to P8 pin 2 and P9 pin 3. The wire can be wire wrapped to make a reliable and neet joint. At most of the sites for connection there is a solder pad;

if not scrape off the green solder resist coating and make a quick clean joint directly to the track.

6. Prepare a new front panel to accommodate the components of your choice, we suggest a 3-position toggle switch FPS1 (300-1200-9600 baud) and a 25-way female "D" connector, mounted with pin 1 towards the bottom of the panel. Wire these to the appropriate places on the socket use for the 145406, UX1. (Use some pin assemblies, and wire wrap to the pins if this is a method which pleases you, otherwise use your own favourite method.)

7. Insert the ICs, connect the peripheral and switch on.

DIL Switch Settings etc

These of course depend on the software in use, but the Interak Tradition is as follows:

VDU Serial Console:

Status Port 00H, Data Port 01H

ie S1 ON-ON-ON-ON-ON-ON-ON-ON

Printer (if the VDU is serial):

Status Port 02H, Data Port 03H

ie S1 ON-ON-ON-ON-ON-OFF-ON-ON

Printer (if the VDU is memory mapped):

Status Port 06H, Data Port 07H

ie S1 ON-ON-ON-ON-OFF-ON-ON-ON

The selection of the baud rate and the settings of S2 depend on the device to which the card will be connected. A very good start for S2 in case of doubt is to leave all switches in S2 "OFF" (in which case you can remove the entire switch pack S2.)

Parts List for Modification

4	3k3 Resistors
3	Decoupling Capacitors (if fitted)
3	1N4ØØX (if fitted)
1	145Ø6 RS-232 Driver Receiver
1	74LS33 Quad Open Collector NOR gate
1	14 Pin DIL socket
1	16 Pin DIL socket
1	Replacement Front Panel
1	25-Way "D" connector and screws
	Prestripped Wire

Software

If you are writing your own program to drive the modified card you may care to read the following. A UART connected in this way is about as simple a serial interface as you are going to get, so should not present a very difficult task even for a beginner.

1. Preliminaries

First define in your program the port address which you have chosen for your serial interface (ie the address to which the switch S1 is set).

Call this the "Status Port".

Define the "Data Port" as being 1 + "Status Port"

Define "Data Available, DAV" as Ø1ØØØØØØ binary (4Ø hex.)

Define "Transmit Buffer Empty, TBMT" as 1ØØØØØØØ binary (8Ø hex.)

Initialise any pointers, variables, etc. (eg set a pointer to a vacant area in memory to act as a buffer for input, or point to a block of data in memory for output).

2. Serial Input.

(i) Input a byte from the Status Port.
(ii) Test DAV; if it is "Ø" go back to (i), otherwise go on to (iii).
(iii) Input a byte (data byte) from the Data Port.
(iv) Process the data byte, eg load a memory location with it, and move a pointer to the next free location in memory.
(v) If there is more data expected go back to (i).

3. Serial Output.

(i) Load a byte (data byte) from memory.
(ii) Input a byte from the Status Port.
(iii) Test TBMT; if it is "Ø" go back to (ii), otherwise go on to (iv).
(iv) Output the data byte to the Data Port.
(v) If there is more data to send, increment the memory pointer and go back to (i).

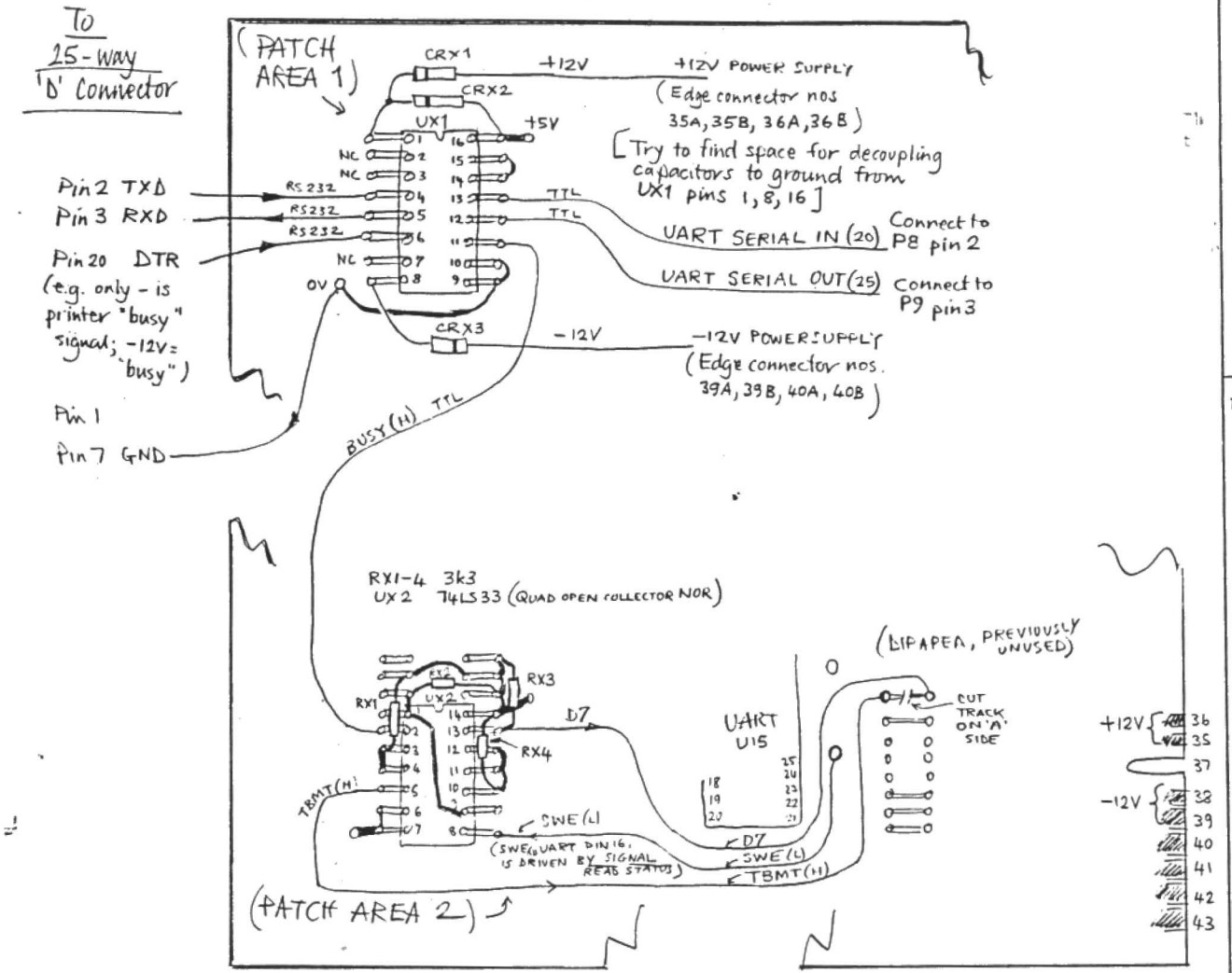
You can see from the above routines that first a check of the UART status is made then data is output or input as appropriate. An alternative approach is to separate the status check from the data output or input. This makes it easier to arrange things so that the processor does not waste too much time idling round in a loop - a quick check of the status of the UART can be done, and data output or input only if the UART is ready, if not some useful other work can be done by the microprocessor.

A further sophistication which should be considered is to add a "time out" in the serial routines. The idea of this is to prevent the routine getting into a never ending loop, waiting say for data from a serial device which has no data to send. We suggest that initially you should write the simplest software, and then add any sophistications you require only when you find you need them.

D M Parkins, B Eng (Hons)

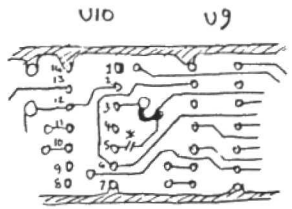
Greenbank Electronics

UX1 = 145406 RS232 DRIVER/RECEIVER
CRX1-3 = DIODE IN400X SERIES, PROTECTION DIODE



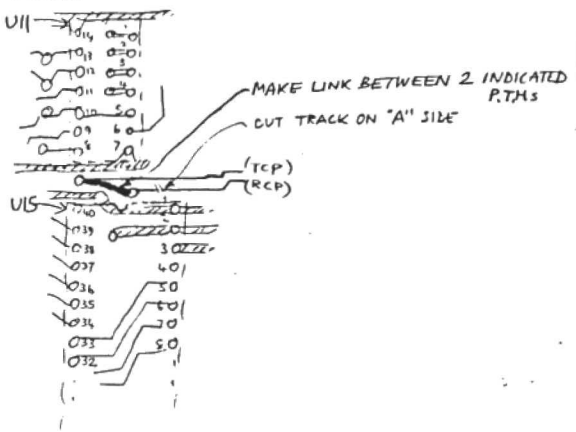
PARTIAL VIEWS ON "B" SIDE OF DTI-1 CARD. SHOWING POSITION AND WIRING OF MODIFICATION COMPONENTS.

PARTIAL VIEW ON 'A' SIDE SHOWING TRACK CUT IN THE AREA OF U9 AND U10, AND POSITION OF LINK TO BE MADE



*ISOLATE PIN 5 OF U10 AND CONNECT TO PIN 3 ON "A" SIDE OF BOARD

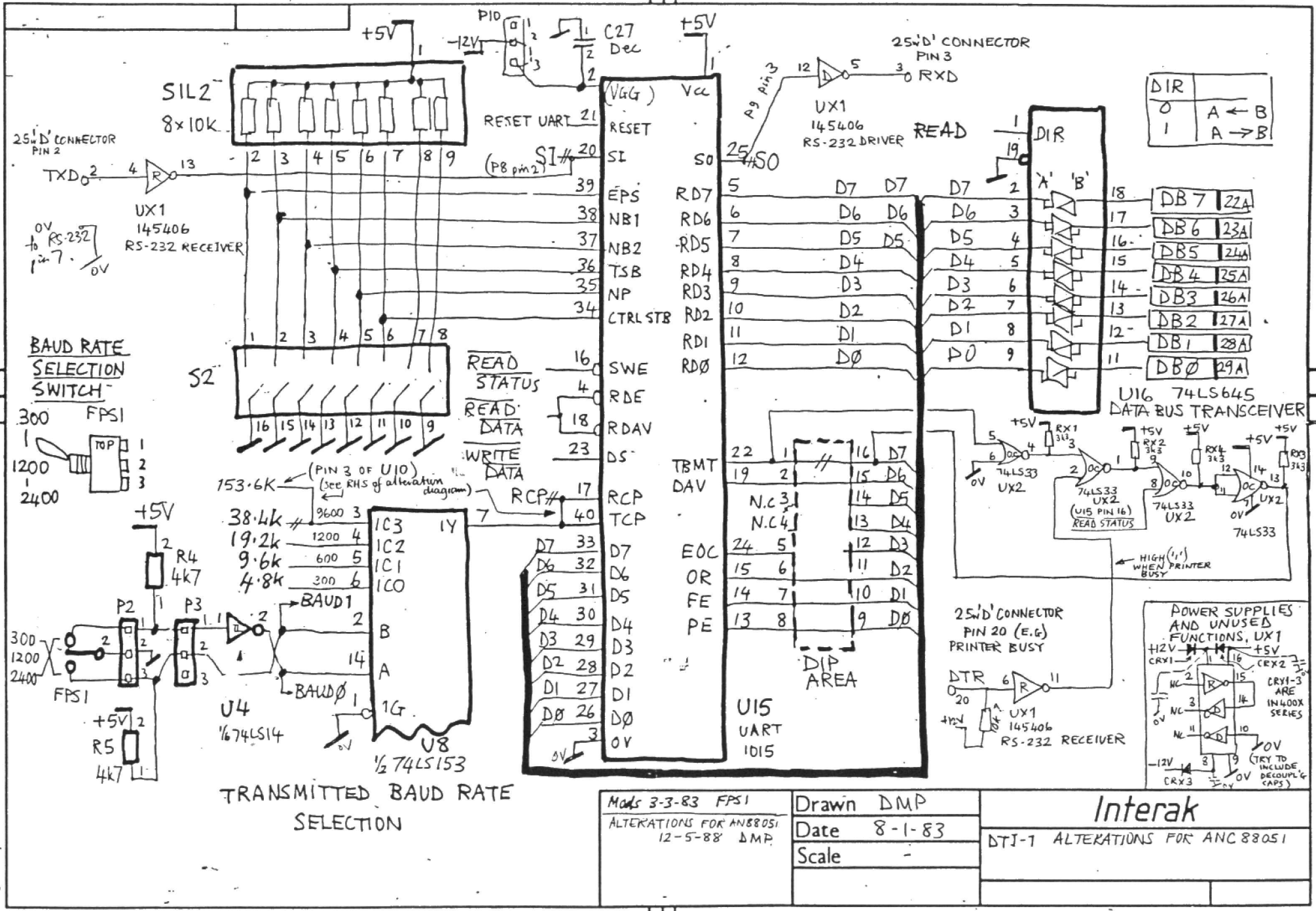
PARTIAL VIEW ON 'A' SIDE SHOWING TRACK CUT NEAR PINS 1 AND 40 OF UART U15, AND POSITION OF LINK TO BE MADE



Drawn	DMP
Date	12-5-88
Scale	-

Greenbank Electronics

DTI-1 ALTERATIONS FOR ANC88051
LAYOUT OF MODIFICATIONS



Mods 3-3-83 FPS1	Drawn DMP
ALTERATIONS FOR AN88051 12-5-88 DMP	Date 8-1-83
	Scale -

Interak

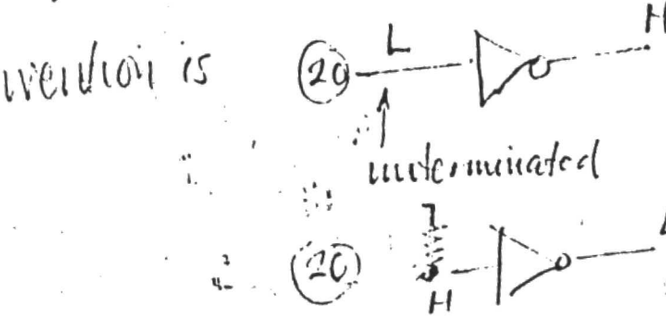
DTI-1 ALTERATIONS FOR ANC88051

Information from Charlie B. 14/6/88

The modified DTI-1 (to Simple serial i/f)

Must have +volts (e.g 5V, 12V) on pin 20 of the 'D' connector if no "busy" signal is present. This connects to pin 6 on the 145406 which sinks low if unconnected

Convention is



Also for economy (if appropriate, ie if only one baud rate required) remove 2nd '393 and associated '153 instead fit header to jumper frequency (for e.g 9600 baud) across '153

Say all above in notes.